BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

2154

PTO/SB/21 (08-03)
Approved for use through 08/30/2003. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/646,983
Filing Date	08/21/2003
First Named Inventor	Hannel
Art Unit	2154
Examiner Name	
Attorney Docket Number	I004-P03079US

Total Number of Pages in This Submission: 24

ENCLOSURES *(Check all that apply)*

- [ ] Fee Transmittal Form
  - [ ] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [x] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Response to Missing Parts/ Incomplete Application
  - [ ] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [x] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] After Allowance communication to Technology Center (TC)
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [x] Other Enclosure(s) (please Identify below):
  Credit Card Form

Remarks

Petition to Make Special under 37 CFR 1.102(d) - Infringement

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	SoCal IP Law Group
Signature	
Date	05-06-04

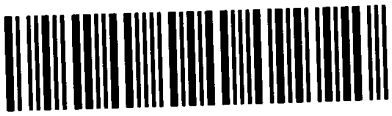
CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Fax Number:

Typed or printed name	Douglas S. Kirk		
Signature		Date	05-06-04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	10/646,983	Confirmation No. 1148
Appl. No. Bar Code	:		
Applicant	:	Hannel	
Filed	:	08/21/2003	
TC/A.U.	:	2154	
Examiner	:		
Docket No.	:	I004-P03079US	
Customer No.	:	33356	
Customer No. Bar Code	:		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

05/11/2004 STEUMEL1 00000151 10646983
01 FC:1460 130.00 OP

Petition to Make Special under 37 CFR 1.102(d) -- Infringement

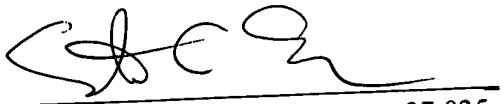
The Commissioner is hereby petitioned to make special the above-identified application.

1. There is an infringing device or product actually on the market or method in use.
2. A rigid comparison of the alleged infringing device, product, or method with the claims of the application has been made, and that, in my opinion, some of the claims are unquestionably infringed; and
3. I have made or caused to be made a careful and thorough search of the prior art or have a good knowledge of the pertinent prior art, and an IDS is enclosed with one copy of each of the references deemed most closely related to the subject matter encompassed by the claims if said references are not already of record.
4. The fee under 37 CFR 1.17(h) is enclosed.

Please contact the undersigned if there are any questions, or if this petition is not in a

form to be granted.

Respectfully submitted,



Steven C. Sereboff, Reg. No. 37,035

Date: May 3, 2004

SoCal IP Law Group
310 N. Westlake Blvd., Suite 120
Westlake Village, CA 91362
Telephone: 805/230-1350
Facsimile: 805/230-1355
email: info@socalip.com

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	10/646,983	Confirmation No. 1148
Appl. No. Bar Code	:		
Applicant	:	Hannel	
Filed	:	08/21/2003	
TC/A.U.	:	2154	
Docket No.	:	I004-P03079US	
Customer No.	:	33356	
Customer No. Bar Code	:		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Information Disclosure Statement

Dear Sir:

General Matters

X	Copies of the references listed on the attached form PTO/SB/08A are enclosed. Copies of US patents and patent applications are not enclosed.

X	37 CFR 1.97(b)(3)	Before the mailing of a first Office action on the merits.

Deposit Account Authorization

Please apply any charges or credits to Deposit Account No. 501524.

Respectfully submitted,

Date: May 5, 2004

Steven C. Sereboff, Reg. No. 37,035

SoCal IP Law Group
310 N. Westlake Blvd., Suite 120
Westlake Village, CA 91362
Telephone: 805/230-1350 x22
Facsimile: 805/230-1355
e-mail: info@socalip.com

PTO/SB/08A (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 2

Complete if Known	
Application Number	10/646,983
Filing Date	08/21/2003
First Named Inventor	Hannel
Art Unit	2154
Examiner Name	
Attorney Docket Number	I004-P03079US

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 6,625,689			
		US- 6,421,730			
		US- 6,401,117			
		US- 6,157,955			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

PTO/SB/08B (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 2

Complete if Known	
Application Number	10/646,983
Filing Date	08/21/2003
First Named Inventor	Hannel
Art Unit	2154
Examiner Name	
Attorney Docket Number	I004-P03079US

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		The Undercover Report: JBoss 3, by Scott Crawford, Version 1.0, 4 June 2003, 16 pages.	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

The Undercover Report: JBoss™ 3

by Scott Crawford

Version 1.0

4 June 2003

Undercover Reports are based on actual tests run on different application server products, and significant care has been taken to ensure that a consistent approach is used that will reveal meaningful results. However, as the reports are provided free of charge, no warranty is made nor any guarantees made as to the accuracy of the report.

For updates, additions, and corrections, please see:

www.scottcrawford.com/undercover

Table of Contents

I. Introduction

A. Concepts

Undercover Reports are based upon investigation of how Enterprise JavaBean (EJB) application servers manage their EJB components. The data presented are derived from tests run using *Undercover EJB Components*, that is, components which are written specifically to monitor how they are being managed. When these components are deployed into a J2EE-compatible application server product, and placed under load by a test client, they reveal information about the workings of the product. Issues such as instance pooling, passivation, and caching of EJB components can be explored using this technique.

Undercover components are intended to answer simple questions of bean-management strategy. Typical of the questions that they hope to answer is "Does application server X pool the instances of session beans it uses to respond to clients, or does it create a new bean for each client?" They may further help with broad conclusions about whether the number of beans used will be affected by the degree of processing performed by a bean or the size of its client load.

However, it is a mistake to read the data with more precision than they are intended to provide. A test that results in 5 bean instances may result in 45 bean instances when repeated. Those looking to draw precise scientific measurements will be disappointed for several reasons. First, timing factors in the tests are non-deterministic and will often vary the exact results. Second, the undercover components are themselves artificial as they are fewer, smaller, and more eccentric than the beans in a real J2EE application. Finally, the hardware on which the tests are run - an ordinary laptop computer - does not resemble that which would be used for a production J2EE system.

A proper introduction to the technique is given in the separate document, *Introduction to Undercover Components*, which is available from www.scottcrawford.com/undercover. A few of the most important points are repeated briefly here as well.

This technique is not useful for performance benchmarking. The products tested use different virtual machines, different wire protocols, different memory configuration, and all have their own unique characteristics. No attempt has been made to provide a "level playing field". Therefore it is neither fair nor useful to compare the performance of one product to the performance of another product on the same test. It is sometimes useful to see how the same product will perform differently in response to two similar tests; these are the only performance comparisons that appear in Undercover Reports.

Default configuration has been used unless otherwise specified on a specific result. All of the products are configurable, and the management of EJBs will be strongly affected by configuration and tuning. Therefore these reports should never be taken as evidence

that a product can not behave in a desired fashion; rather, they reveal "out of the box" behaviour.

Only one specific version of each product is tested for each J2EE release; the target version for the tests is the first one identified by its vendor as being fully production capable. Since some vendors come to market with a full production release much more rapidly than others, this means that products that came out at very different times can be compared in these reports. It is also important to remember that any problematic behaviours identified in the reports may have been improved by a patch or minor release which occurs very shortly after the first production release.

These reports focus on tests based on RMI calls and JMS messages sent remotely to Enterprise JavaBeans. This narrow focus allows for a deep analysis but it also avoids areas, such as incoming HTTP requests, in which products are likely to have strengths.

B. Details of Tested Product and Platform

Vendor:	An open-source community centred around The JBoss Group
Product:	JBoss Application Server
Version:	3.0.0
Certification:	Not officially certified. JBoss claims EJB 2.0 and J2EE 1.3.
Released:	Summer 2002
URL:	http://www.jboss.org/
Platform:	Windows 2000 Professional
Hardware:	IBM ThinkPad T20 with Pentium III and 384 MB RAM

C. Document History

Date	J2EE version	JBoss version	Document version	Comments
4 June 2003	1.3	3.0	1.0	First published version

Please Note:

The results presented on the following pages are accompanied by only a brief explanation giving the purpose of each test. This avoids repetition of background material in all of the product-specific reports. For details on the mechanics of how the testing works, please refer to the separate document *Introduction to Undercover Components*. To ensure sensible interpretation of the results, it is best to read this report in conjunction with the document *Conclusions and Comparisons from the 1.3 Results*.

D. Special Note About RMI and JBoss 3.0.0

Most undercover tests are based on the principle that many test clients will make almost simultaneous RMI calls to the EJB server. The JBoss version tested, however, exhibited a particular problem with this type of load: when the number of clients in the burst exceeded 100, the test would usually fail with multiple RMI ConnectExceptions.

Discussion in the JBoss Forums established that this was a known problem expected to be fixed in a later minor release, possibly version 3.0.3. However, the policy for undercover testing (see section A) is that the first production-capable version for each J2EE release is put through the tests. In order to be fair to all vendors, this policy has been adhered to, and the following results are based on JBoss version 3.0.0. This is why the results show test cases of up to only 100 or sometimes 200 clients.

The decision to release JBoss 3.0 with such a significant problem was made by the JBoss community, and it seems reasonable that they should take responsibility for it. It is worth noting, however, that as on open-source product, JBoss would have certain migrating advantages. Any customers in a grave situation with this problem would have the option of looking at the product source code and possibly patching it themselves. This consideration is useful for understanding why it is sometimes difficult to compare JBoss on a like-for-like basis with more commercial alternatives.

II. Session Bean Tests

A. *Pooling of Stateless Session Beans*

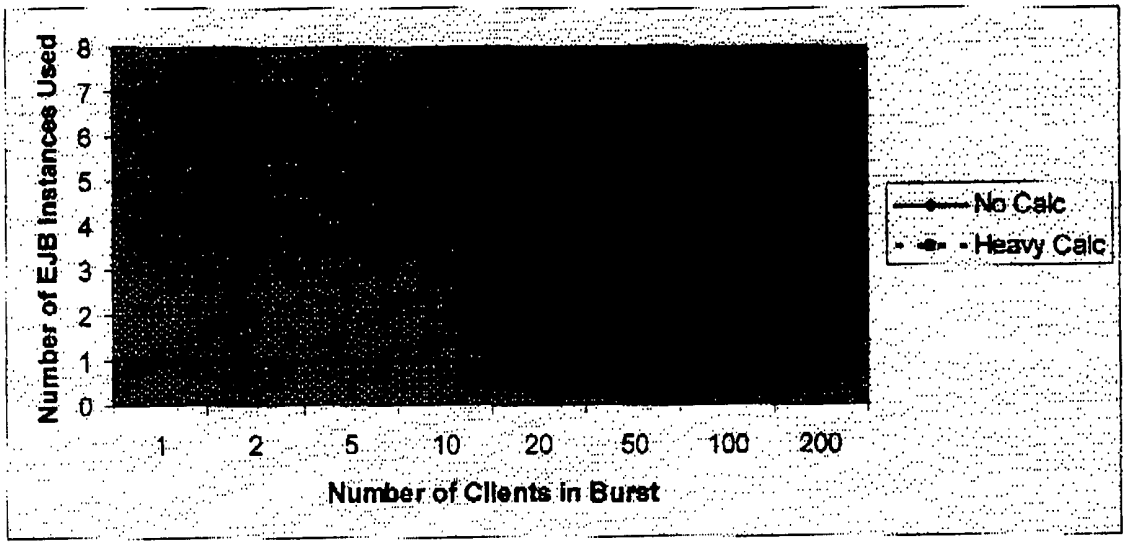


Fig 1. Results of Stateless Session Bean Pooling Test

The pooling tests are designed to determine whether an application server pools stateless session bean instances, and if possible, identify factors which influence the size of the pool. Each of the points on the graph represents a test where a burst of client requests has been made on an individual session bean. At the left end are tests where only a single client request has been made; at the right end are tests where 200 client requests have been made in a single burst. The Y-axis reveals the number of instances of the session bean which were used to handle each load.

The JBoss 3 results show that a high degree of instance pooling is occurring. For a lightweight session bean which does very little processing ("No Calc"), all of the client loads can be handled by a single instance.

The "Heavy Calc" test run shows the same test for a session bean which performs significant processing – each bean invocation results in one million square root calculations. Here the number of beans required is larger than in the no calculation case, but there is still significant pooling occurring.

So far as they have succeeded, these results are similar to those of other J2EE 1.3 application servers. Unfortunately it has not been possible to get results up to the standard test load of 5000 clients. (For explanation see section I.D).

B. Stateful Session Beans

B1. Instances of Stateful Session Beans

Because stateful session beans hold state which is specific to a client, their instances are not pooled. A pooling test such as that shown in section A (above) on JBoss 3 stateful beans shows that the number of bean instances used is always equal to the number of clients in the burst; the graph shows $y = x$.

In theory, there is some latitude for application server vendors to use an implementation which would not always ensure one bean instance for each client; however, all of the J2EE 1.3 products tested to date show this result for stateful beans.

B2. Passivation of Stateful Session Beans

One area where the products are not all alike is passivation. The potentially large number of stateful bean instances can cause a shortage of memory; consequently application servers are allowed to passivate instances to disk. However, it is not easy to observe this behaviour on certain products, and it is possible their designers have decided not to implement passivation.

Unfortunately, due to the problems of RMI load testing on JBoss 3.0.0 (see explanation in section I.D) it has not been possible to do a meaningful test for passivation in JBoss. However, discussions in the JBoss forums suggest that the product does passivate stateful beans in practice.

C. Stateless and Stateful Session Performance

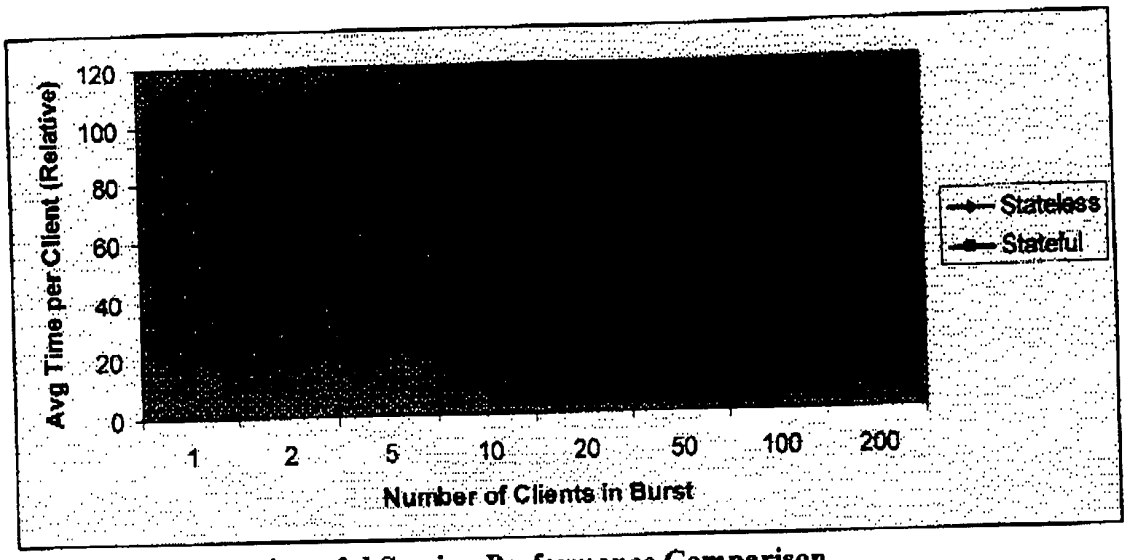


Fig 2. Stateless vs. Stateful Session Performance Comparison

The performance comparison shows a series of tests which have been timed. The total time from when the clients are started until they are all serviced is measured and then divided by the number of clients. This average time provides a measure of how efficiently the application server is handling the load. Because undercover testing is not a performance benchmark, the Y-axis is expressed in relative units, where the average time for the first stateless test has been assigned an arbitrary reference value of 100.

Two series of tests are shown: one where the undercover session bean is declared to be stateless and one where it is declared stateful. It has already been demonstrated that the stateful tests will involve the creation of a larger number of bean instances than the stateless tests. The key question for this test to answer is whether that difference will have a significant impact on performance.

In both cases there is an encouraging decrease in average time as the number of clients becomes significant. Possibly due to the number of bean instances created, however, the stateful tests perform somewhat worse than the stateless ones. Unfortunately it is not possible to see if this difference continues into higher loads because of the problems with RMI load testing (see section I.D).

III. Message-Driven Bean Tests

Tests confirm that JBoss 3 pools Message Driven Bean (MDB) instances in a manner similar to the pooling of stateless session bean instances. For example, a burst of 10 clients that send messages to the same JMS topic will be serviced by a single MDB instance.

Unfortunately, tests succeed reliably only when the number of clients in the burst is 20 or lower. Bursts of 50 JMS clients will often fail. Unlike the problems with RMI load testing, this problem under JMS load is quite common among J2EE 1.3 application servers. As with all of them, it is possible that configuration and tuning will help.

It is also worth remembering that the current undercover MDB implementation opens and closes a new connection on behalf of each client publisher; a more typical load scenario would be that where a few publishers each publish many messages. This and other known problems are discussed in *Introduction to Undercover Components.*

IV. Entity Bean Tests

A. *Bean Instances and Primary Keys*

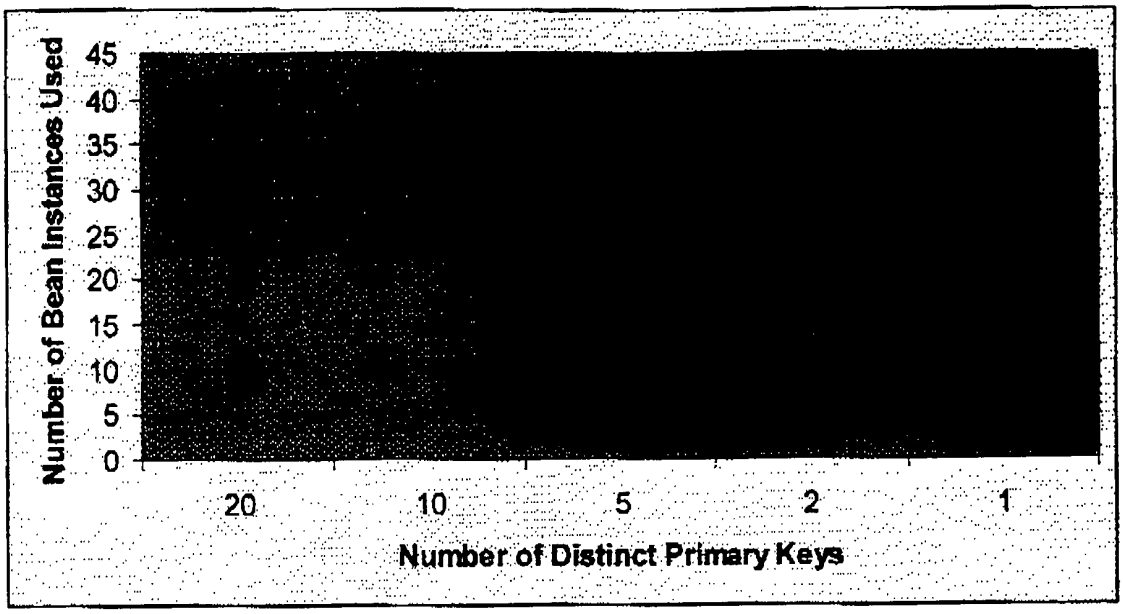


Fig 4. Entity Bean Contention Tests – 20 Clients in Each Burst

N.B. Due to RMI load problems (see section I.D) the standard 1000 clients have not been used in this test series. Instead a load of 20 clients in each burst was used.

The contention tests reveal what the determining factor is for how many entity bean instances are used. In some products, there is a clear relationship maintained between primary keys and bean instances, whereas in others, the bean instances are generic tools that are created in response to total client load. Each of the tests shown above involves a burst of 20 clients which are contending over a variable number of primary keys. The test on the far right shows a case where all 20 clients are targeting the same primary key.

The results demonstrate that JBoss 3 is maintaining a relationship between bean instances and primary keys. In fact, there is a consistent result that the number of bean instances created during a given test is exactly the number of distinct primary keys plus the number of clients. This suggests that, once in the in the ready state, beans remain there to be re-used by those seeking the same primary key. The impact from the number of clients is harder to understand. One possibility is that each client is causing its own bean instance to be created into the pooled state even though it ought to be possible to re-use them.

B. Commit Options A/B/C

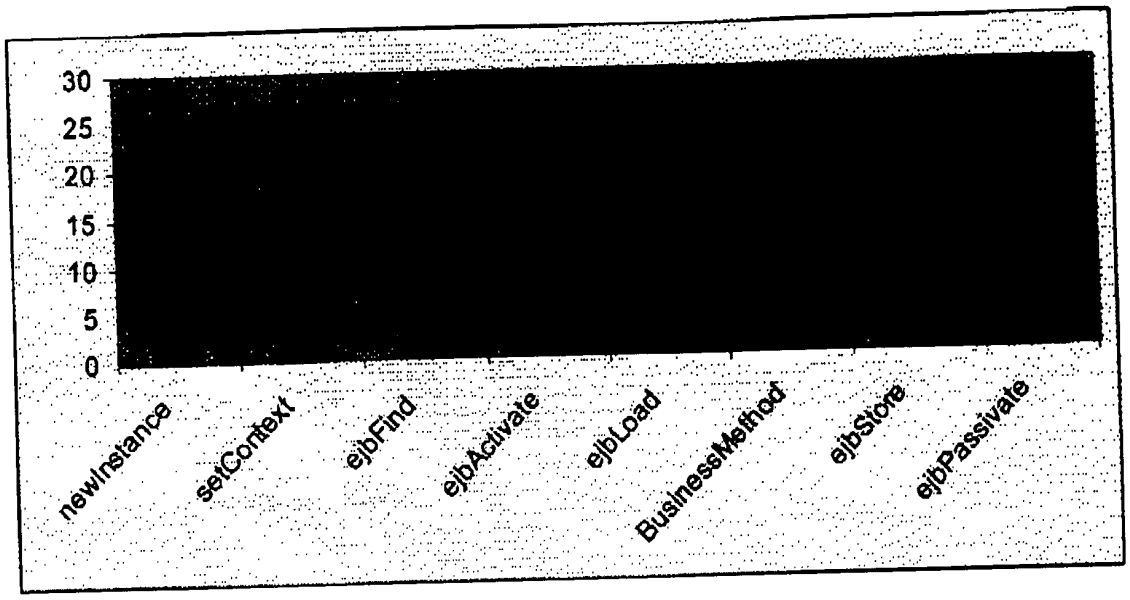


Fig 5. Commit Option Test – 20 Clients on 5 Primary Keys

N.B. Due to RMI load problems (see section I.D) the standard 100 clients against 20 keys have not been used in this test series. Instead a load of 20 clients against 5 keys was used.

The commit option test examines a single burst of client activity and all of the callback notifications that result. In this case, 20 clients were started and each one followed this procedure:

- Look up the home interface for the undercover entity bean.
- Call findByPrimaryKey() using an integer primary key determined by calculating the client number modulo 5.
- Call the business method of the entity bean.

The resulting counts of each kind of callback notification reveal which commit option is being used by default, and may also spot unusual activity.

The result above shows that JBoss is, by default, using an Option A commit. This is evident since the number of ejbLoad notifications is much lower than the number of business method calls. This indicates that once the server has loaded the state of the bean instance from the database, it assumes that that state is valid from then on and does not refresh it later when the bean is re-used.

The "standardjboss.xml" configuration file has many switches relating to commit options, and an examination of it suggests that Option A will not be the default in all entity bean

cases. Since the undercover entity bean is a bean-managed persistence (BMP) bean running on a standalone (non-clustered) server, it has defaulted to Option A behaviour. However, several other varieties of entity beans, including all of those deployed in a cluster, will default to Option B behaviour.

Incidentally, the unusually high number of newInstance and setContext calls is consistent with the suggestion (in section A) that JBoss is creating a new entity instance in the pooled state to handle each client. It is hard to see why the instances are not being re-used.

V. General Comments

A. *Documentation, Examples, and Ease-of-Use*

Subjective Grade for Documentation, Examples, and Ease-of-Use: F
(possible grades are A-F)

Documentation is a problem for open-source projects, and the JBoss Group points out that developers working without pay can not be motivated to produce it. Therefore JBoss has a program of documentation-for-sale and provides a free Getting Started guide. Unfortunately, in the period immediately after the 3.0.0 release, no current documentation was available, either for-sale or free. The only obtainable Getting Started guide was for the previous 2.x version, and it was disappointing.

Even finding what documentation and examples did exist was difficult since they were not included in the main product download, and the JBoss web site was confusing for this purpose. Significant time was wasted in tracking down three separate downloads. Once downloaded, it became clear that a lot of trial and error would be required. The experience is simply not comparable to that of downloading one of the commercial products which has made a serious effort to help you get started.

B. *Key Points on JBoss 3*

JBoss is different. Its proponents point out several advantages, most importantly that it's free. Unlike the J2EE Reference Implementation, which is also free, JBoss is licensed for use in production systems. Those with a difficult support problem and the time required can study the server source code. Finally, the undercover tests confirm that there is a real EJB container there, since it is capable of executing the tests.

But the tests reveal real problems too. The RMI load (section I.D) and extra entity instances (section IV) issues are significant. The effort to execute the tests also showed a problem with true J2EE compatibility. Undercover tests make use of the "application client container" facility which is required by J2EE 1.3. This resolves the EJB references used by the client. Unlike all other J2EE 1.3 products tested, however, JBoss has not implemented this facility and it was necessary to change the clients to use JNDI names directly. Given that JBoss has not been subjected to the rigors of J2EE compatibility testing, it would be surprising if this were the only omission.

Those who are sensitive to issues of strict compatibility or quality assurance in release procedures should note the particular disadvantages of JBoss as well as its advantages. Those with tight budgets, their own J2EE expertise, and the extra time necessary to benefit from the selling points of JBoss, will be glad to have this distinct alternative.

C. Summary and History of Observations

The following chart shows a summary of the undercover observations about JBoss. Since version 3 is the first one that has been subject to undercover testing, no historical data are available. For comparison with other application server products, please see the document *Conclusions and Comparisons from the 1.3 Results.*

	JBoss 3.0
Pools stateless sessions	Yes
Passivates stateful sessions	? Likely
Pools MDB	Yes
Entity instances used in Find/Call	Keys + Clients
Default entity commit option (BMP)	A (B for some cases)

- END -

Send comments, questions, or corrections to undercover@scottcrawford.com